(12) UK Patent Application (19) GB (11) 2 308 688 (13) A

(43) Date of A Publication 02.07.1997

(21) Application No 9624244.1

(22) Date of Filing 21.11.1996

(30) Priority Data
(31) 08580108 (32) 29.12.1995 (33) US

(51) INT CL[6]
G06F 12/14 1/00

(52) UK CL (Edition O)
G4A AAP

(56) Documents Cited
None

(58) Field of Search
UK CL (Edition O) G4A AAP
INT CL[6] G06F
ONLINE:WPI

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

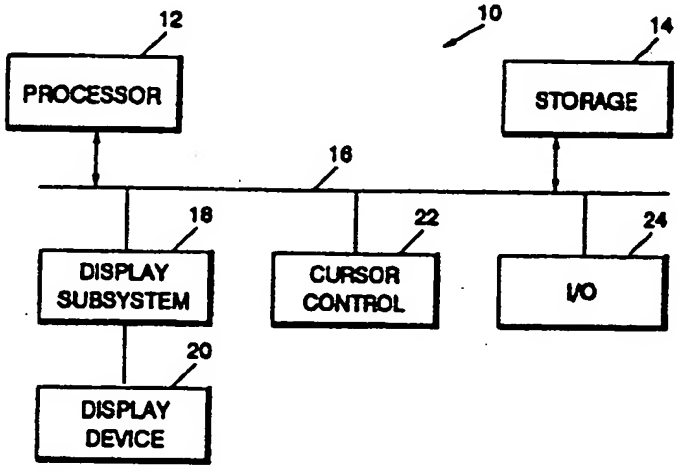
(72) Inventor(s)
Messaoud Benantar
George Robert Blakeley III
George Prentice Copeland
Anthony J Nadalin

(74) Agent and/or Address for Service
G A Bailey
IBM United Kingdom Limited,
Intellectual Property Department, Mail Point 110,
Hursley Park, WINCHESTER, Hampshire, SO21 2JN,
United Kingdom

(54) Controlling access to objects in an information handling system

(57) A system, method and article of manufacture for controlling access to objects in an information handling system employing object oriented technology, includes one or more processors (12), a memory system (14), one or more I/O controllers (24) each controlling one or more I/O devices (18,20,22), a bus (16) connecting the processors (12), the memory system (14) and the I/O controllers (24), an operating system controlling operation of the processors (12), the memory system (14) and the I/O controllers (24), and an object oriented control means which includes a BeforeAfter metaclass having a control element which is used to control access to objects. A secure metaclass is defined to be a subclass of the BeforeAfter metaclass. The secure metaclass overrides the Before method to perform authorisation checking prior to method dispatching. This controls access to protected classes which are subclasses of the secure metaclass.

FIG. 1



GB 2 308 688 A

FIG. 1

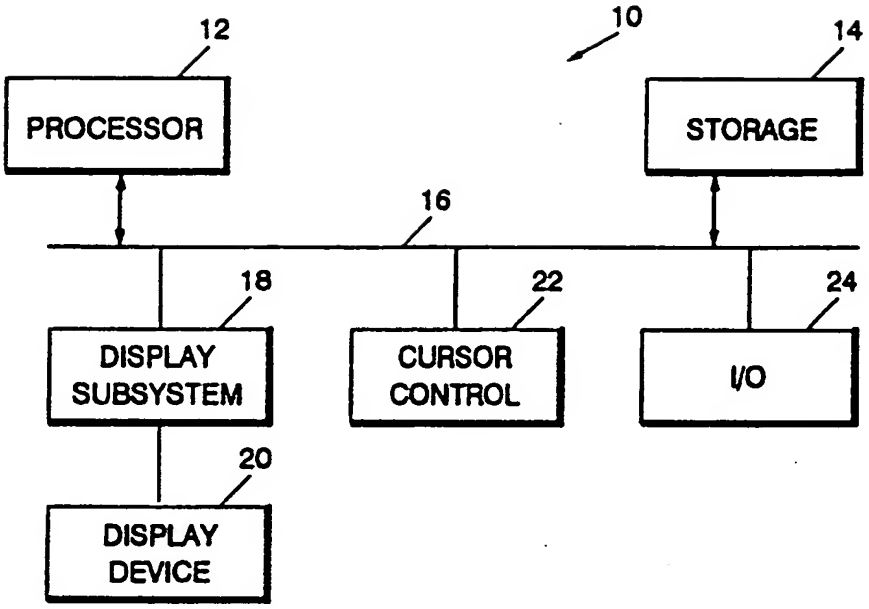

12
10
14
PROCESSOR
STORAGE
16
18
22
24
DISPLAY SUBSYSTEM
CURSOR CONTROL
I/O
20
DISPLAY DEVICE

FIG. 2

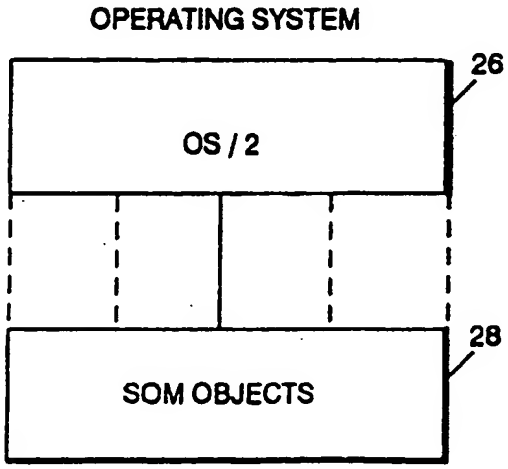
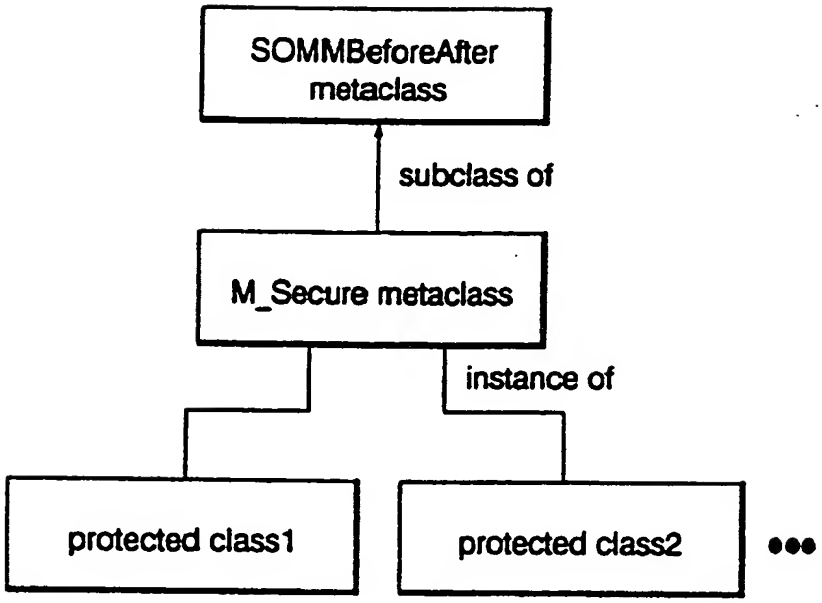

OPERATING SYSTEM
26
OS / 2
28
SOM OBJECTS

FIG. 3


SOMMBeforeAfter metaclass
subclass of
M_Secure metaclass
instance of
protected class1
protected class2
•••

2308688

CONTROLLING ACCESS TO OBJECTS IN AN INFORMATION HANDLING SYSTEM

The present invention relates to information handling systems, methods and articles of manufacture, and more particularly to information handling systems, methods and articles of manufacture for protecting objects by access control in an object oriented system.

There is a need to protect access to objects in an object oriented system.

Traditionally, resource managers represent the control point for enabling access authorisation checking. On a request for accessing a resource in a data processing system, its resource manager takes control, and if the underlying resource is protected, the access checking process is triggered, after which the requested access is either granted or stopped from proceeding. For instance, on a file opened, resulting from a print instruction of a user's program, the file system invokes an authorisation checking routine that verifies that the user has the right for writing into the corresponding file.

In object oriented systems, with resources being object instances, there is no such traditional concept of resource managers that would take control on every attempt to invoke a method, or send a message, to a protected object.

The problem, as it relates to a certain class of object oriented systems, is finding a way of automatically triggering access checking on a method invocation on a protected object. It is required that any solution be safe from client tampering, that is, the client should not be able to effect or disrupt automatic triggering of authorisation checking.

In the prior art there are many techniques for improving object oriented programming systems (OOPS).

The following are examples of the prior art.

U.S. Patent 4,525,780 teaches a data processing system having a memory organised into objects, where the system uses a protection technique to prevent unauthorised access to objects by users which are identified by a subject number identifying the user, a process for executing a user's procedure and the type of system operation to be performed by the user's procedure. An access control list is maintained to control access to objects based on subject identifier.

Although the patent generally deals with methods for protecting access to objects in object oriented programs, the patent does not teach nor suggest using a metaclass in an object oriented system to control access to objects as is taught and claimed herein with reference with the present invention.

U.S. Patent 5,136,712 teaches an object based operating system for a multitasking computer system including means for making an object private to a process. Access to a private object is controlled by an access control list.

Although the patent generally deals with methods for protecting access to objects in object oriented programs, the patent does not teach nor suggest using a metaclass in an object oriented system to control access to objects as is taught and claimed herein with reference with the present invention.

U.S. Patent 5,265,221 teaches an access control mechanism for granting, revoking, and denying authorisation

to objects using a system of verbs, parameters, attributes and functions.

Although the patent generally deals with methods for protecting access to objects in object oriented programs, the patent does not teach nor suggest using a metaclass in an object oriented system to control access to objects as is taught and claimed herein with reference with the present invention.

U.S. Patent 5,297,283 and US Patent 5,321,841 appear to teach the same system as US Patent 5,136,712 discussed above.

Although the patent generally deals with methods for protecting access to objects in object oriented programs, the patent does not teach nor suggest using a metaclass in an object oriented system to control access to objects as is taught and claimed herein with reference with the present invention.

U.S. Patent 5,093,914 generally teaches a method used by a digital computer in controlling execution of an object oriented program to effect a defined action, for example, stopping the program when a specified virtual function is invoked on a specified object during execution of the program.

Although the patent generally deals with methods for protecting access to objects in object oriented programs, the patent does not teach nor suggest using a metaclass in an object oriented system to control access to objects as is taught and claimed herein with reference with the present invention.

U.S. Patent 5,343,554 teaches a computing system in which problems are solved by a process which includes

creating objects of first and second classes wherein the second class objects are externally invocable and where the externally invocable sub-class objects include an indication of an internally invocable sub-class object and executing the class of objects wherein one externally invocable sub-object invokes the internally invocable sub-object and a new object of the first class is generated in response to the results.

Although the patent generally deals with methods for protecting access to objects in object oriented programs, the patent does not teach nor suggest using a metaclass in an object oriented system to control access to objects as is taught and claimed herein with reference with the present invention.

An overall goal in object oriented programming systems is the development of program objects which can be reused easily.

The importance of binary code over source code increases with the degree of reuse. A main purpose of object oriented technology is code reuse. For object oriented technology to achieve large scale success, binary code reuse is essential. As the state of the art moves towards applications built from object formats which can be tailored and plugged into each other, binary code is critical.

The Object Management Group is defining a set of interfaces for object system services named Common Object Services.

It is an object of the present invention to protect objects in an object oriented system by controlling access to objects.

According to the first aspect of the present invention, there is provided a method for controlling access to objects in an information handling system employing object oriented technology, comprising the steps of defining a secure metaclass; invoking a first class as a metaclass of the secure metaclass; and controlling access to one or more protected classes which are subclasses of the secure metaclass in response to values returned on invoking the first class.

According to the second aspect of the present invention, there is provided a computer readable medium comprising means for controlling access to objects in an information handling system employing object oriented technology, the means comprising means for defining a secure metaclass; means for invoking a first class as a metaclass of the secure metaclass; and means for controlling access to one or more protected classes which are subclasses of the secure metaclass in response to values returned on invoking the first class.

According to the third aspect of the present invention, there is provided an information handling system, employing oriented technology, for controlling access to objects comprising one or more processors; a memory system; one or more I/O controllers each controlling one or more I/O devices; a bus connecting the processors, the memory system and the I/O controllers; an operating system controlling operation of the processors, the memory system and the I/O controllers; means for defining a secure metaclass; means for invoking a first class as a metaclass of the secure metaclass; and means for controlling access to one or more protected classes which are subclasses of the secure metaclass in response to values returned on invoking the first class.

It is an advantage of the present invention that object security can be readily achieved by creation of a secure metaclass which performs authorisation checking prior to method dispatch.

The invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of a system for executing the method according to the present invention.

Figure 2 is a block diagram showing the operating system platform and system object model program supporting the present invention.

Figure 3 is an object diagram of the preferred embodiment of the present invention.

Referring now to Figures 1 and 2, an information handling system 10 and an operating system environment for the present invention will be described.

Information handling system 10 may be a graphics work station or the like with a very powerful processor 12, a storage system 14, a system bus 16, display sub-system 18 controlling a display device 20, a cursor control device 22, and an I/O controller 24, all connected by system bus 16. The information handling system 10, shown in Figure 1 may be operated by a commercially available well known multitasking operating system such as OS/26 (OS/2 is a registered trademark of International Business Machines Corporation). Among many tasks which the OS/2 operating system 26 controls on operating information handling system 10, is execution of a program SOMObjects 28, which is a commercially available product of International Business Machines Corporation.

The method and article of manufacture of the present invention may be incorporated in the SOMObjects program. As used herein, SOM means system object model.

An important aspect of any object oriented system is making the object secure, so that object/method granularity access control can be enforced. This involves checking access control in a before method.

The present invention may be efficiently embodied in Distributed SOM (DSOM) (across address space ORB) and Security SOM (SSOM).

The way the user specifies security is to specify the following constraints when searching for a class object:

name: secure

value: A boolean indicating whether the object is protected by access control checking.

name: ACL

value: If secure==TRUE, an access control list.

When granularity is per method, enforcement can be done either per method invocation or when a reference is first obtained (the capability approach). The present invention provides enforcement on method invocation.

A class is also an object whose class is a metaclass. A class object, then, being an instance of its metaclass, responds to the methods that its metaclass defines. A metaclass, like any other class, can be subclassed and its methods be overridden. All metaclasses are ultimately derived from the root metaclass.

Referring now to the object diagram of Figure 3, the preferred embodiment of the present invention will be described. One run time metaclass is a BeforeAfter metaclass. When a class is created from the BeforeAfter metaclass 32, two methods are transparently executed on every invocation of method M on an object of class C. One method executes before the method is invoked, the Before method and the other method, the After method, executes following the execution of method M. The Before method and the After method are defined by the BeforeAfter metaclass 32. These two methods can be overridden in a metaclass 34 derived from the BeforeAfter metaclass to allow for a special function to be performed.

On invocation, the Before method returns a boolean value (1 or 0). If this value is TRUE (1), then normal processing occurs, that is, the execution of the primary method M, followed by the execution of the After method. If the boolean value is not TRUE (0), neither the primary method M nor the After method are executed. Thus, the BeforeAfter metaclass 32 is the key to controlling method execution on protected objects 36, 38.

A secure metaclass 34 (M_Secure) is defined to be a subclass of BeforeAfter metaclass 32. In M_Secure, the Before method of metaclass 32 is overridden to perform authorisation checking prior to dispatch of the Before method from metaclass 32. A protected class 36, 38 is a class whose metaclass is M_Secure 34. M_Secure overrides BeforeMethod to perform the following function:

```
BeforeMethod(M_secure    Self,

             Environment    *ev,

             Object         object,
```

```
        ID              primaryMethod,

        va_list         ap)

    {

retrieve privileges or capabilities of calling principal;

retrieve the authorisation attributes of object and
MethodId;

    return access_allowed(...,
                principal_security_attributes,

                primaryMethod_security_attributes,

                object_security_attributes)
```

The boolean value returned by BeforeMethod corresponds to the access decision made in access_allowed of the run-time security component of DSOM. Thus, the primary method is executed only when the access checking method allows for it.

Note that the solution cannot be tampered with by the client application as the access decision is performed on the server side during the attempt to dispatch the method.

CLAIMS

1. A method for controlling access to objects in an information handling system employing object oriented technology, comprising the steps of:

defining a secure metaclass (34);

invoking a first class (32) as a metaclass of the secure metaclass (34); and

controlling access to one or more protected classes (36,38) which are subclasses of the secure metaclass (34) in response to values returned on invoking the first class (32).

2. A method according to claim 1, wherein the step of controlling access further comprises the steps of:

overriding dispatch of any method invoked by the invoking step; and

checking authorisation for access to an object in a protected class (36,38) prior to method dispatching.

3. A method according to claim 2, wherein the step of controlling access further comprises the step of:

dispatching an invoked method if the checking authorisation step indicates authorised access to a protected class (36,38).

4. A method according to claim 1, 2 or 3, wherein the invoking step further comprises the steps of:

invoking a before method on the secure metaclass (34);

testing a value returned by the before method;

if the value returned is true, proceeding with normal processing; and

if the value returned is not true, inhibiting further processing.

5. A computer readable medium comprising means for controlling access to objects in an information handling system employing object oriented technology, the means comprising:

means for defining a secure metaclass (34);

means for invoking a first class as a metaclass (32) of the secure metaclass (34); and

means for controlling access to one or more protected classes (36,38) which are subclasses of the secure metaclass (34) in response to values returned on invoking the first class (32).

6. A computer readable medium, according to claim 5, wherein said means for controlling access further comprises:

means for overriding dispatch of any method invoked by the invoking means; and

means for checking authorisation for access to an object in a protected class (36,38) prior to method dispatching.

7. A computer readable medium, according to claim 6, wherein said means for controlling access further comprises:

means for dispatching an invoked method if the checking authorisation means indicates authorised access to a protected class (36,38).

8. A computer readable medium, according to claim 5, 6 o 7, wherein said means for invoking further comprises:

means for invoking a before method on the secure metaclass (34);

means for testing a value returned by the before method;

means for proceeding with normal processing if the value returned is true; and

means for inhibiting further processing if the value returned is not true.

9. An information handling system, employing oriented technology, for controlling access to objects comprising:

one or more processors (12);

a memory system (14);

one or more I/O controllers (24) each controlling one or more I/O devices (18,20,22);

a bus (16) connecting the processors (12), the memory system (14) and the I/O controllers (24);

an operating system (26) controlling operation of the processors (12), the memory system (14) and the I/O controllers (24);

means for defining a secure metaclass (34);

means for invoking a first class (32) as a metaclass of the secure metaclass (34); and

means for controlling access to one or more protected classes (36,38) which are subclasses of the secure metaclass (34) in response to values returned on invoking the first class (32).

10. An information handling system, according to claim 9, wherein said means for controlling access further comprises:

means for overriding dispatch of any method invoked by the invoking means;

means for checking authorisation for access to an object in a protected class (36,38) prior to method dispatching; and

means for dispatching an invoked method if the checking authorisation means indicates authorised access to a protected class (36,38).

The Patent Office

45

Application No:	GB 9624244.1	**Examiner:**	Mike Davis
Claims searched:	1-10	**Date of search:**	20 January 1997

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): G4A (AAP)

Int Cl (Ed.6): G06F

Other: Online: WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
	None	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.